



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/602,196	06/23/2003	Marc Solsona	50072.0059US01/NC28812US	1004

826 7590 01/16/2007

ALSTON & BIRD LLP
BANK OF AMERICA PLAZA
101 SOUTH TRYON STREET, SUITE 4000
CHARLOTTE, NC 28280-4000

EXAMINER

GERGISO, TECHANE

ART UNIT

PAPER NUMBER

2137

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	01/16/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No. 10/602,196	Applicant(s) SOLSONA ET AL.	
	Examiner Techane J. Gergiso <i>T. G</i>	Art Unit 2137	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 24 October 2006.
- 2a) ☒ This action is FINAL. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) _____ is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-29 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This is a Final Office Action in response to the applicant's amendment filed on October 24, 2006.
2. The applicant amended claims 1, 8, 16, 18, 21-22, 27 and 29.
3. Claims 1-29 have been examined.
4. Claims 1-29 are pending.

Response to Arguments

5. Applicant's arguments filed October 24, 2006 have been fully considered but they are not persuasive.
6. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

7. Applicant's argument; Page 8: Third Paragraph

The applicant argues, "Kavsan discloses cryptographic service software embodied that electronically communicates with a standard operating system of a personal computer. The cryptographic service software **performs cryptographic service in the kernel space of the operating system**. See Kavsan's Abstract, and Figure. Thus, unlike the claimed invention, Kavsan does not disclose or suggest **services on the kernel space or other components of the operating system**. "

The examiner disagrees with the applicant's argument because consider and interpret **"performing cryptographic service in the kernel space of the operating system"** is substantially equivalent in function and scope to **"cryptographic services on the kernel space or other components of the operating system."**

8. Applicant's argument; Page 8: Fourth Paragraph

The applicant argues, "It is clear that **Kavsan does not determine integrity data for an operating system binary**. Kavsan's cryptographic service software **performs cryptographic services at the kernel space (but not on the kernel space - or any component of the operating system, including an operating system binary)**. Kavsan's cryptographic service software also discloses that its **algorithms may be used to encrypt signals at the driver level,**"

The examiner again disagrees with the applicant's argument because the examiner consider and interpret **"performing cryptographic services at the kernel space"** is substantially equivalent in function and scope to **"performing cryptographic services on the kernel space including component of the operating system."**

9. Applicant's argument; Page 9: Third Paragraph

The applicant argues, "Alexey does not disclose or suggest **modifying the kernel with the integrity.**"

The examiner again disagrees with the applicant's argument because Alexey teaches in page 3 (second and third paragraph) that "In certain cases, it is very important to have access to cryptographic services in the kernel mode. For instance, file and disk encryption products and

Art Unit: 2137

implementations of Virtual Private Network (VPN) concept usually include kernel mode components which make extensive use of cryptographic functions, such as encryption, hashing, and random bits generation. For such a component, cryptographic service providers residing in the user.... Therefore, the F-secure Kernel Mode Cryptographic Driver, whose high performance API functions can be directly called from other kernel mode drivers, may bring considerable value to software vendors developing real-time data security products for Microsoft Windows NT, Windows 2000, and Windows XP Operating Systems.”

10. Therefore, the applicant’s argument and remark are not persuasive to overcome the prior arts of record and they do not place the claims in condition for allowance. According independent claims 1, 8, 18, 22 and 29 are not in condition for allowance. Dependant claims 2-7, 9-17, 19-21 and 23-28, depending directly or indirectly from their respective independent claims are also not placed in condition for allowance.

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

12. Claims 1-5 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kavsan (US Pat. No.: 6,412, 069) in view of Alexey Kirichenko (F-Secure Kernel Mode Cryptographic

Driver (Microsoft® Windows™ NT/2000/XP) FIPS 140-2 Validation Security Policy Created:
December 2001, Module version: 1.1)

As per claim 1:

Kavsan teaches a method for protecting an operating system, comprising:

determining integrity data for an operating system binary, wherein the integrity data enables detection of a modification to the operating system binary (Column 2: lines 10-24; Column 2, lines 61-67; Column 3: lines 5-15, 20-27); and
modifying a kernel with the integrity data, wherein the kernel is operable to employ the integrity data to detect the modification to the operating system binary (Column 3: lines 35-52; lines 54-65).

Kavsan does not explicitly disclose determining integrity data and detection of a modification to the operating system binary. Alexey in analogous art, however, disclose determining integrity data and detection of a modification to the operating system binary (Page 7: Paragraph 3). Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to modify the system disclosed by Kavsan to include determining integrity data and detection of a modification to the operating system binary. This modification would have been obvious because a person having ordinary skill in the art would have been motivated to do so to provide a Kernel Mode Cryptographic Driver, whose high performance API functions can be directly called from other kernel mode drivers, may bring considerable value to software vendors developing real-time data security products for Microsoft

Windows NT, Windows 2000, and Windows XP Operating Systems as suggested by Alexey in (Page 3: Paragraph 2).

As per claim 2:

Alexey discloses a method, wherein the integrity data further comprises at least one of a digital signature, and a hash associated with the operating system binary (Page 7: Paragraph 4).

As per claim 3:

Alexey discloses a method, wherein the hash further comprises at least one a message digest, and a Secure Hash Algorithm (SHA) (Page 7: Paragraph 4).

As per claim 4:

Alexey discloses a method, wherein the modifying the kernel further comprises:
storing the integrity data in a data store (Page 11: Paragraph 3); and
embedding the data store into the kernel (Page 18: Paragraph 1-3);

As per claim 5:

Alexey discloses a method, wherein embedding the data store in the kernel further comprises at least one of digitally signing the data store, and encrypting the data store (Page 7: Paragraph 4).

Art Unit: 2137

13. Claims 6-7 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kavsan (US Pat. No.: 6,412, 069) in view of Alexey Kirichenko (F-Secure Kernel Mode Cryptographic Driver (Microsoft® Windows™ NT/2000/XP) FIPS 140-2 Validation Security Policy Created: December 2001, Module version: 1.1) in further view of Pham et al. (US Pub No.: 2004/0078568).

As per claim 6:

Alexey teaches generating an operating system image based in part on the modified kernel and the operating system user level binary (Page 11: Paragraph 3).

the operating system image comprises at least one of creating an archive file, a compressed file, and a Cabinet (CAB) file.

Kavsan and Alexey do not explicitly disclose the operating system image comprises at least one of creating an archive file, a compressed file, and a Cabinet (CAB) file. Pham et al. in analogous art, however, disclose the operating system image comprises at least one of creating an archive file, a compressed file, and a Cabinet (CAB) file (Figure 5B: 42; Figure 12: 388). Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to modify the system disclosed by Kavsan and Alexey to include the operating system image comprises at least one of creating an archive file, a compressed file, and a Cabinet (CAB) file. This modification would have been obvious because a person having ordinary skill in the art would have been motivated to do so to provide an efficient mechanism

Art Unit: 2137

for reliably securing persistent data in a manner eminently subject to cooperative management and control within a security domain as suggested by Pham et al. in (Page 2: 0012).

As per claim 7:

Kavsan discloses a method, wherein the operating system binary further comprises at least one of an OS user level binary, and the kernel (Figure 1: Application Space ; Kernel Space).

14. Claims 8-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Eun et al. (WO 01/80482 A1) in view of Alexey Kirichenko (F-Secure Kernel Mode Cryptographic Driver (Microsoft® Windows™ NT/2000/XP) FIPS 140-2 Validation Security Policy Created: December 2001, Module version: 1.1)

As per claim 8:

Eun et al. disclose a method for protecting an operating system, comprising;
generating a first integrity data for an operating system binary (Page 5: lines 11-20; lines 28-34; Page 6: lines 4-11);
modifying an operating system kernel with the first integrity data (Page 8: lines);
receiving a request associated with the operating system binary (Page 8: lines 15-22);
retrieving the first integrity data associated with the operating system binary (Figure 3: 312, 314 318);
determining if the first integrity data indicates tampering of the operating system binary (Figure 3: 310 308 306); and

performing a tamper detection action if the first integrity data indicates tampering of the operating system binary (Figure 3: 310 308 306).

Eun et al. do not explicitly disclose modifying an operating system kernel. Alexey in analogous art, however, disclose modifying an operating system kernel (Page 7: Paragraph 3). Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to modify the system disclosed by Eun et al. to include modifying an operating system kernel. This modification would have been obvious because a person having ordinary skill in the art would have been motivated to do so to provide a Kernel Mode Cryptographic Driver, whose high performance API functions can be directly called from other kernel mode drivers, may bring considerable value to software vendors developing real-time data security products for Microsoft Windows NT, Windows 2000, and Windows XP Operating Systems as suggested by Alexey in (Page 3: Paragraph 2).

As per claim 9:

Eun et al. disclose a method, wherein receiving the request further comprises receiving at least one of a read action, an execute operation, and an install request (Figure 8: 702).

As per claim 10:

Alexey discloses a method, wherein performing the tamper detection action further comprises at least one of providing a tamper detection message, and quarantining the operating system binary (Page 7: Paragraph 4).

As per claim 11:

Alexey discloses a method, wherein the first integrity data further comprises at least one of a digital signature, and a hash associated with the operating system binary (Page 7: Paragraph 4).

As per claim 12:

Alexey discloses a method, wherein the hash further comprises at least one a message digest, and a Secure Hash Algorithm (SHA) (Page 7: Paragraph 4).

As per claim 13:

Alexey discloses a method, wherein modifying the operating system kernel with the first integrity data further comprises storing the first integrity data in at least one of a database, a file, and a program (Page 12: Paragraph 3).

As per claim 14:

Alexey discloses a method, wherein modifying the operating system kernel further comprises associating the first integrity data with the operating system kernel (Page 11: 12).

As per claim 15:

Alexey discloses a method, wherein associating the first integrity data with the operating system kernel further comprises digitally signing the first integrity data with a digital key associated with the operating system kernel (Page 11: 12).

As per claim 16:

Eun et al. disclose a method, wherein determining if the first integrity data indicates tampering of the operating system binary further comprises:

determining a second integrity data for the operating system binary (Page 2: lines 15-27;

Abstract; Page 7: lines 15-20);

determining if the first integrity data is substantially different from the second integrity data (Page 6: lines 25-36); Page 7: lines 15-20); and

indicating tampering of the operating system binary if the first integrity data is substantially different from the second integrity data (Page 13: lines 16-33).

As per claim 17:

Eun et al. disclose a method, wherein determining if the first integrity data is substantially different from the second integrity data further comprises comparing the second integrity data to the first integrity data (Page 2: lines 15-27; Abstract; Page 7: lines 15-20).

As per claim 18:

Eun et al. disclose a method for protecting an operating system, comprising:

receiving a request associated with an operating system binary (Page 8: lines 15-22);

retrieving integrity data associated with the operating system binary (Figure 3: 312, 314 318); and
performing a tamper detection action if the integrity data indicates tampering of the operating system binary (Figure 3: 310 308 306).

Eun et al. do not explicitly disclose modifying an operating system kernel. Alexey in analogous art, however, disclose modifying an operating system kernel (Page 7: Paragraph 3). Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to modify the system disclosed by Eun et al. to include modifying an operating system kernel. This modification would have been obvious because a person having ordinary skill in the art would have been motivated to do so to provide a Kernel Mode Cryptographic Driver, whose high performance API functions can be directly called from other kernel mode drivers, may bring considerable value to software vendors developing real-time data security products for Microsoft Windows NT, Windows 2000, and Windows XP Operating Systems as suggested by Alexey in (Page 3: Paragraph 2).

As per claim 19:

Eun et al. disclose a method, wherein receiving the request further comprises receiving at least one of a read action, an execute operation, and an install request (Figure 8: 702).

As per claim 20:

Alexey discloses a method, wherein performing the tamper detection action further comprises at least one of providing a tamper detection message, and quarantining the operating system binary (Page 7: Paragraph 4).

As per claim 21:

Eun et al. disclose a method, wherein determining if the integrity data indicates tampering of the operating system binary further comprises:

determining another integrity data for the operating system binary (Page 2: lines 15-27;

Abstract; Page 7: lines 15-20);

determining if the other integrity data is substantially different from the retrieved integrity data (Page 6: lines 25-36); Page 7: lines 15-20); and

indicating tampering of the operating system binary if the other integrity data is substantially different from the retrieved integrity data (Page 13: lines 16-33).

15. Claims 22-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Eun et al. (WO 01/80482 A1) in view of Pham et al. (US Pub No.: 2004/0078568).

As per claim 22:

Eun et al. disclose a computer-readable medium having computer-executable components for protecting an operating system, comprising:

a data store configured to receive and store a first integrity data, wherein the first integrity data is for an operating system binary (Figure 3: 312, 314, 316, 318); and

receiving a request to examine an operating system binary (Page 6: lines 5-11; Page 7: 4-22);
retrieving the first integrity data for the operating system binary (Page 8: lines 11-22);
determining if the first integrity data indicates tampering of the operating system binary (Page 11: lines 15-33).

Eun et al. do not explicitly disclose a tamper detection component, coupled to the data store, that is arranged to perform actions, and performing a tamper detection action if the first integrity data indicates tampering of the operating system binary. Pham et al. in analogous art, however, disclose a tamper detection component, coupled to the data store, that is arranged to perform actions, and performing a tamper detection action if the first integrity data indicates tampering of the operating system binary (Figure 10B: 302; Figure 12B: 382). Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to modify the system disclosed by Eun et al. to include a tamper detection component, coupled to the data store, that is arranged to perform actions, and performing a tamper detection action if the first integrity data indicates tampering of the operating system binary. This modification would have been obvious because a person having ordinary skill in the art would have been motivated to do so to provide an efficient mechanism for reliably securing persistent data in a manner eminently subject to cooperative management and control within a security domain as suggested by Pham et al. in (Page 2: 0012).

As per claim 23:

Pham et al. a computer-readable medium, wherein the computer-executable components are associated with an operating system kernel (Figure 5A: 42).

As per claim 24:

Pham et al. a computer-readable medium, wherein performing the tamper detection action further comprises at least one of providing a tamper detection message, and quarantining the operating system binary (Figure 12B: 382).

As per claim 25:

Eun et al. a computer-readable medium, wherein the first integrity data further comprises at least one of a digital signature, and a hash associated with the operating system binary (Figure 3: 304).

As per claim 26:

Eun et al. a computer-readable medium, wherein the operating system binary further comprises at least one of an OS user level binary, and a kernel (Figure 2: User Level, Kernel Level).

As per claim 27:

Pham et al. a computer-readable medium, wherein determining if the first integrity data indicates tampering of the operating system binary further comprises:

determining a second integrity data for the operating system binary (Figure 5B: 156);

determining if the first integrity data is substantially different from the second integrity data (Figure 10B: 298), and
indicating tampering of the operating system binary if the first integrity data is substantially different from the second integrity data (Figure 10B: lines 302).

As per claim 28:

Eun et al. a computer-readable medium, wherein the second integrity data further comprises at least one of a digital signature, and a hash associated with the operating system binary (Figure 3: 304).

As per claim 29:

Eun et al. disclose an apparatus for protecting an operating system, comprising: means for receiving a request to examine an operating system binary;

means for retrieving a first integrity data for the operating system binary (Page 8: lines 11-22);

means for determining a second integrity data for the operating system binary (Page 6: lines 25-36); Page 7: lines 15-20); and

Eun et al. do not explicitly disclose means for determining if the first integrity data is substantially different from the second integrity data, and if the first integrity data is substantially different from the second integrity data, a means for performing a tamper detection action. Pham et al. in analogous art, however, disclose means for determining if the first integrity data is

substantially different from the second integrity data, and if the first integrity data is substantially different from the second integrity data, a means for performing a tamper detection action. (Figure 10B: 302; Figure 12B: 382).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to modify the system disclosed by Eun et al. to include means for determining if the first integrity data is substantially different from the second integrity data, and if the first integrity data is substantially different from the second integrity data, a means for performing a tamper detection action. . This modification would have been obvious because a person having ordinary skill in the art would have been motivated to do so to provide an efficient mechanism for reliably securing persistent data in a manner eminently subject to cooperative management and control within a security domain as suggested by Pham et al. in (Page 2: 0012).

Conclusion

16. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

See the notice of reference cited in form PTO-892 for additional prior art

17. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the date of this final action.

Contact Information

18. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Techane J. Gergiso whose telephone number is (571) 272-3784 and fax number is **(571) 273-3784**. The examiner can normally be reached on 9:00am - 6:00pm. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Emmanuel Moise can be reached on (571) 272-3865. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications

Art Unit: 2137

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

T. G.

Techane Gergiso

Patent Examiner

Art Unit 2137

January 6, 2007



EMMANUEL L. MOISE
SUPERVISORY PATENT EXAMINER